Machine Learning
https://doi.org/10.1007/s10994-023-06472-0

Word embeddings for retrieving tabular data from research publications

Alberto Berenguer[1] · **Jose-Norberto Mazón**[1] · **David Tomás**[1]

Received: 14 March 2023 / Revised: 6 August 2023 / Accepted: 17 October 2023

Abstract
Scientists face challenges when finding datasets related to their research problems due to the limitations of current dataset search engines. Existing tools for searching research datasets rely on publication content or metadata, do not considering the data contained in the publication in the form of tables. Moreover, scientists require more elaborate inputs and functionalities to retrieve different parts of an article, such as data presented in tables, based on their search purposes. Therefore, this paper proposes a novel approach to retrieve relevant tabular datasets from publications. The input of our system is a research problem stated as an abstract from a scientific paper, and the output is a set of relevant tables from publications that are related to the research problem. This approach aims to provide a better solution for scientists to find useful datasets that support them in addressing their research problems. To validate this approach, experiments were conducted using word embedding from different language models to calculate the semantic similarity between abstracts and tables. The results showed that contextual models significantly outperformed non-contextual models, especially when pre-trained with scientific data. Furthermore, the importance of context was found to be crucial for improving the results.

Keywords Research tabular data · Information retrieval · Word embeddings · Text classification

1 Introduction

Scientists frequently come across research datasets while reading articles or conducting publication searches. Specifically, when retrieving different parts of an article based on their search objectives, scientists have different requirements (Hagiwara et al., 2022). These requirements go beyond the capabilities of current one-size-fits-all research search engines

Editors: Dino Ienco, Roberto Interdonato, Pascal Poncelet.

✉ Alberto Berenguer
aberenguer@dlsi.ua.es

[1] Department of Software and Computing Systems, University of Alicante, Carretera San Vicente del Raspeig s/n, 03690 San Vicente del Raspeig, Spain

(Mysore et al., 2023). For instance, when scientists aim at "finding previous research in an unfamiliar field", they often need access to the full-text of a research article, particularly the "Introduction" and "Related work" sections. On the other hand, when "searching for research methods", they rely on information found in the "Materials and Methods" section, as well as data tables within the article (Hagiwara et al., 2022). Additionally, it is worth noting that researchers may not necessarily be interested in the datasets themselves. Instead, they may focus on utilising the dataset descriptions (metadata) found within publications to gain an understanding of the datasets and determine their relevance and utility (Gregory et al., 2020).

Therefore, scientists need to identify appropriate datasets to achieve their research goals, which is a non-trivial task (Paullada et al., 2021). Currently, research dataset search tools face two main issues. Firstly, they heavily rely on publication content, metadata, and dataset metadata rather than leveraging the content of the papers themselves (Mysore et al., 2023). Secondly, these tools predominantly utilise keywords, which may not be suitable for scientists seeking a more nuanced approach that necessitates the ability to input complete phrases explicitly expressing their research problem (Färber & Leisinger, 2021). Therefore, novel retrieval approaches are required to enable scientists to discover datasets associated with publications related to their research problem.

Importantly, as noted by Chen et al. (2019), data requirements are often formulated as text rather than using keywords. Moreover, as pointed out by Färber and Leisinger (2021), "the quality of the dataset search can be considerably improved when using a rich formulation of the research problem in natural language, rather than relying purely on isolated keywords or attributes".

Hence, research dataset search tools should incorporate a mechanism for finding datasets based on the description of a research problem. Interestingly, research article abstracts consistently contain information about the addressed research problem (Kang et al., 2022). Thus, as described by Färber and Leisinger (2021), abstracts from scientific papers are likely to be utilised as inputs for research dataset search tools since they resemble research problem descriptions.

To motivate this fact, the following example is inspired by the illustration proposed in the work by Viswanathan et al. (2023) to demonstrate the advantages of using phrase-based input instead of keyword-based input when searching for research datasets. A keyword-based query to discover datasets could be "campylobacter jejuni sheep abortion", while a phrase-based query based on an abstract could be "campylobacter jejuni is the major cause of sheep abortion and contributes significantly to foodborne illness in the USA [...]". Utilising an abstract better defines the data needs of researchers as it implicitly addresses two requirements related to causality: researchers not only require a dataset on *campylobacter jejuni* (as indicated by keywords), but also as a cause of sheep abortion (i.e., not as a consequence) while considering the impact on human diseases (i.e., not as a cause). This is better described by using an abstract from a research paper.

The objective of this study is to present and validate the research hypothesis that scientists can retrieve tabular datasets associated with an abstract from a scientific publication, which serves as a representation of a research problem. To accomplish this, word embeddings (Turney & Pantel, 2010) are employed to calculate the semantic similarity between an input abstract and the tables indexed from scientific publications. The output is a collection of relevant tables from publications that are pertinent to the research problem. Multiple experiments have been conducted to validate this hypothesis.

Figure 1 provides an overview of the system's architecture and functioning, illustrating its two primary components: corpus processing and cleansing, and abstract classification.

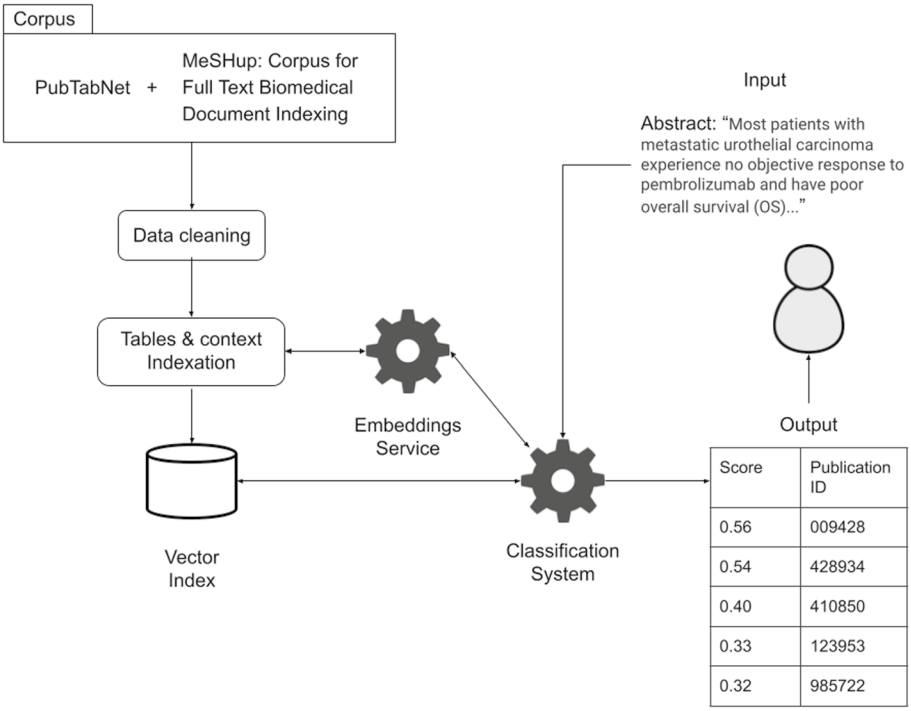


Fig. 1 Overview of the table retrieval approach

The initial component, corpus processing and cleansing, is crucial for data preparation before analysis. The corpus undergoes several preprocessing steps to ensure compatibility with the system's embedding and indexing processes. Once the corpus is processed and cleaned, the system generates word embeddings for each model being evaluated, which are then stored in the vector index. The vector index enables efficient retrieval of relevant documents during the classification process. The second component, abstract classification, is responsible for categorising tables from scientific publications that are relevant to the input abstract. The classification process is based on the similarity between the word embeddings of the abstract and the word embeddings generated for each table in the dataset.

The remainder of this article is structured as follows: Sect. 2 presents related work; Sect. 3 describes the approach based on word embeddings for retrieving tabular datasets from scientific publications; Sect. 4 outlines the experimentation conducted to validate this approach; Sect. 5 contains the discussion of the obtained results; finally, Sect. 6 presents the conclusions and future work.

2 Related work

The approach proposed relies on the use of word embeddings to provide a semantically rich vector representation of the tables in the dataset. Word embeddings are dense vectors that represent the meaning of a word as a point in a semantic space. These continuous representations can be used in downstream natural language processing (NLP) tasks, such

as text classification (Lilleberg et al., 2015) and question answering (Shih et al., 2016). From a linguistic perspective, they represent the distributional meaning of words (Turney & Pantel, 2010), that is, the meaning that a word assumes in a specific text regardless of the meaning it may have in the dictionary (Harris, 1954; Firth, 1957). Thus, similar representations are learnt from words appearing in similar contexts. In the scientific literature, they are differentiated from traditional semantic vectors, were the meaning of a word is represented as a sparse vector with the weight of its components calculated using measures such as TF-IDF (Salton & Buckley, 1988). Examples of this type of word representation are Word2vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014).

These word embedding techniques build a global vocabulary using unique words in the documents, assigning a single representation for each word and ignoring that they can have different meanings or senses in different contexts. They are considered as *static* representations unable to capture the different senses of a word. On the other hand, recent contextual word embeddings (Devlin et al., 2019) are able to capture the different meanings of polysemous words, since each vector represents not a word but a sense. In this way, each word is represented with different word embeddings, one for each context in which the word can occur. During the training process, contextual word embeddings are generated taking into consideration the surrounding words, that is, the sequence of words in the sentence or text span in which a word appears. Examples of this type of representation are ELMo (Peters et al., 2018), ULMFit (Howard & Ruder, 2018), and BERT (Devlin et al., 2019), among others (Liu et al., 2020).

Another notable distinction between these contextual models and their static predecessors is the utilisation of subword units rather than full words to represent the vocabulary. Word embeddings are constructed based on the specific set of tokens available in the corpus utilised for vector creation. When encountering an out-of-vocabulary word in a new text, word-based models do not provide a representation for it in the semantic space, resulting in the token being considered as *unknown*. To handle the large vocabularies commonly found in natural language corpora, BERT employs the WordPiece subword segmentation algorithm (Wu et al., 2016). This approach initialises the vocabulary with individual characters in the language and then progressively incorporates the most frequent combinations of symbols into the vocabulary. Consequently, subwords have their own representation in the semantic space, enabling the assignment of a representation to previously unknown words by combining the vectors of their underlying subword units.

Current contextual language models use the widely adopted Transformer architecture (Vaswani et al., 2017). Transformers are neural network architectures that rely on attention mechanisms, thereby dispensing with the need for convolutional and recurrent networks. The original architecture consists of a multi-head self-attention mechanism combined with an encoder-decoder structure.

In the field of NLP, the introduction of BERT marked a significant breakthrough in language models (Devlin et al., 2019). This architecture introduced a bidirectional encoder that captures information from both the left and right contexts of a word during the training phase. The model enables transfer learning in NLP tasks, wherein the BERT model initially trained on one dataset (the *pre-trained model*) can be employed to perform similar tasks on another dataset (the *fine-tuned model*). Current NLP state-of-the-art systems leverage the semantic relationships identified by Transformers as a starting point to solve problems rather than constructing models from scratch, subsequently fine-tuning them on relatively smaller datasets for specific tasks.

In recent years, these models have been adapted to the task of table retrieval. The objective of table retrieval, also known as table search, is to provide a ranked list of tables that are

considered relevant to a given search query (Zhang & Balog, 2018). Depending on the nature of the query, table retrieval can be categorised as either keyword-based or table-based search (Zhang & Balog, 2020). In the former, a set of keywords constitutes the query, similar to traditional search engines like Google. In the latter, the query itself is a table, and the aim is to compute a similarity score between the input table and candidate tables.

In this study, the focus is on retrieving tabular data that is relevant to a specific scientific contribution, represented by its abstract. Therefore, the relevant approaches for this work are those that fall into the keyword-based search task.

One of the first approaches to this task was described in Cafarella et al. (2008), which implemented keyword table search on top of an existing web search engine. This research was further expanded upon in Cafarella et al. (2009), where the authors introduced a system called OCTOPUS that extended the previous method with a reranking mechanism that took attribute co-occurrences into account.

The work presented in Zhang and Balog (2018) employed semantic matching between queries and tables, representing them in multiple semantic spaces and introducing various similarity measures for matching these semantic representations. Both queries and tables were represented using word embeddings (Word2vec) and graph embeddings, among other techniques. They utilised early and late fusion patterns as matching methods between queries and tables.

In Deng et al. (2019), non-contextual word embeddings were used to transform tabular data into vector spaces. The authors considered various table elements such as captions, column headings, and cells to train these embeddings, which were employed in three table-related tasks: row population, column population, and table retrieval.

In the work by Zhang et al. (2019), word embeddings were trained using Wikipedia as a table corpus. Similarly, the study conducted by Bhagavatula et al. (2013) focused on Wikipedia tables and introduced features related to the connectivity of web pages, including incoming and outgoing links.

Regarding the use of Transformer models for keyword-based table retrieval, Chen et al. (2020) employed a pre-trained version of BERT, leveraging different information available in the table (both textual and numerical) to provide BERT with context, such as title, caption, column headings, and cell values.

In Agarwal et al. (2021), tables were treated as 2D images, and traditional neural approaches to image processing (e.g., CNNs) were employed to handle the data. Another image-based neural representation approach was presented in Du et al. (2021), where an image-based method was combined with a graph-based approach to harness the strengths of both techniques. The authors proposed using the WordNet structure as a graph, representing cell texts (tokens) from the table with their corresponding synsets in WordNet. This approach constructed a graph that captured lexical similarities between text cells.

Although there have been recent proposals of contextual word embedding models trained on tabular data (Herzig et al., 2020; Yin et al., 2020), they primarily focus on answering natural language questions from tables and do not address the retrieval of this type of data.

3 Retrieving scientific datasets from research problem descriptions

The primary objective of this research paper is to evaluate the hypothesis that scientists can effectively retrieve tables from publications based on the given abstract, which serves as a representation of the research problem. To achieve this goal, word embedding representations generated by language models are used to assess the relationship between the

abstract and relevant tables (see Fig. 1). Furthermore, in order to evaluate the significance of the table context in the classification process, keywords that are relevant to the paper are employed as a representation of the context in which the tables occur.

This paper investigates various factors that influence scientists' search for research datasets, including:

- The performance of contextual and non-contextual language models
- The significance of pre-training data
- The influence of contextual information surrounding the table
- The real-world system's performance time

As mentioned earlier, Fig. 1 presents an overview of the proposed approach, encompassing several components that are elaborated upon in the subsequent sections.

3.1 Preprocessing and indexation

Before proceeding with content indexing, it is essential to ensure that the data is cleaned and prepared for processing by the language model. Since the model processes input as a string, the different fields of the papers, namely the abstract, table, and keywords, need to be formatted as strings.

The abstract is already in string format, while the list of keywords is converted into a single string, with each element separated by a blank space. As for the table, it is divided into two primary components. The headers are transformed into a string, with each header text separated by a blank space. Additionally, we obtain a separate string for each column content within the table as shown in Fig. 2.

The data cleaning process involves several steps. First, dots, hyphens, underscores, unprocessed column names, and possible HTML labels from the text are removed. Furthermore, CamelCase words are split and text is converted to lowercase. These cleaning procedures ensure that the data is in a suitable format for further processing.

The subsequent step aims to transform these strings into a format that complies with the specifications of the language model, thereby generating an appropriate input. The input capacity of the model typically limits the maximum number of tokens that can be processed to 256 tokens. Exceeding this limit may result in the loss of crucial information. To prevent such loss, an additional step is required, which involves dividing the content into strings of appropriate token lengths. This not only prevents truncated information loss but also enables parallel processing, which accelerates the calculation of word embeddings.

Once the data is clean and formatted as a set of strings, the next step involves table and context indexing (as depicted in Fig. 1). In the indexing process, each string generated during the preprocessing step is processed by the language model, resulting in the generation of a dense vector with a length ranging from 300 to 1024 dimensions, depending on the specific model used. Three indexes are created in total: one for the keyword embeddings and two for the tables. In the case of tables, the first index stores the embeddings of the headers, while the second index stores the average embedding of each column content.

Subsequently, the generated vector is normalised and stored in an index using the Faiss library.[1] This library is designed for efficient similarity search and clustering of dense

[1] https://faiss.ai/.

Abstract: "Campylobacter jejuni clone SA is the major cause of sheep abortion and contributes significantly to foodborne illnesses in the United States…"

Keywords: ["Animals", "Campylobacter Infections", "Campylobacter jejuni", "Cattle Diseases", "Pest Control", "Prevalence", "Prospective Studies", "Retrospective Studies","Starlings", "United States"]

state	samples_tested	campylobacter_positive_%	C_jejuni_isolates_%	clone_SA_isolate_%
Lowa	400	87.5	70.1	2.9
Texas	250	84.2	68.0	8.3
Kansas	360	63.8	79.3	6.9

Preprocessing process

Abstract: "campylobacter jejuni clone sa is the major cause of sheep abortion and contributes significantly to foodborne illnesses in the united states"

Keywords: "animals campylobacter infections campylobacter jejuni cattle diseases pest control prevalence prospective studies retrospective studies starlings united states"

headers: "state samples tested campylobacter positive % c jejuni isolates % clone sa isolate %"
column 1: "lowa texas kansas"
column 2: "400 250 360"
column 2: "87.5 84.2 63.8"
column 3: "70.1 68.0 79.3"
column 4: "2.9 8.3 6.9"

Fig. 2 Example of string formatting for different fields of the paper

vectors. Leveraging this library enables convenient and efficient indexing and searching for similar documents.

3.2 Classification

The objective of the classification task in the proposed approach is to identify relevant tables among the indexed tables based on a given abstract. Relevant tables are defined as those that would belong to the article corresponding to the input abstract.

To accomplish this classification task, the semantic similarity between the abstract and each indexed table is assessed using Eq. 1, which yields a similarity value ranging from −1 (indicating the lowest similarity) to 1 (representing very high similarity). This classification is performed for each of the models under evaluation.

$$sim(A, T) = \alpha \cdot sim(A, T_h) + (1 - \alpha) \cdot sim(A, T_c), \tag{1}$$

where A represents the abstract, T denotes the table, T_h corresponds to the embeddings of the table headers, T_c refers to the embeddings of the table content, and α is a parameter that ranges from 0 to 1, indicating the relevance of the headers and content in the final similarity score.

To evaluate the semantic similarity between each part of the table and the abstract, the abstract undergoes the preprocessing steps described earlier. Subsequently, the word embeddings are obtained and normalised. A search is then conducted using Faiss to retrieve the most similar headers and table content.

Once Faiss returns the results with their corresponding similarities, the equation is applied, and the results are sorted in descending order based on the scores. This sorting

process is repeated while varying the value of the α parameter in the equation. Through this iterative process, we obtain a list of the five most similar indexed tables to the input abstract. This information is later utilised to evaluate the accuracy of the system.

3.2.1 Adding contextual information

Although the information provided by metadata may not always be of sufficient quality to ensure accurate searches, certain metadata elements can be valuable as fields for enriching searches with contextual information. Therefore, it is important to evaluate the inclusion of such information in the classification process to determine whether and to what extent it positively impacts the system's performance. In particular, keywords associated with the papers were incorporated in the tables being searched, allowing the classification to consider not only the similarity between the abstract and the table but also the associated keywords.

To integrate the keyword information into Eq. 1, two variations have been proposed. The first approach involves averaging the similarity scores between the abstract-table pair and the abstract-context (table keywords) pair, as shown in Eq. 2. The second approach involves boosting the context by multiplying it with the similarity score, as shown in Eq. 3. In both equations, C represents the context (keywords) embedding.

$$sim(A, T, C) = (\alpha \cdot sim(A, T_h) + (1 - \alpha) \cdot sim(A, T_c)) \cdot \beta + sim(A, C) \cdot (1 - \beta) \quad (2)$$

$$sim(A, T, C) = (\alpha \cdot sim(A, T_h) + (1 - \alpha) \cdot sim(A, T_c)) \cdot sim(A, C) \quad (3)$$

By incorporating these variations, the classification process takes into account both the similarity between the abstract and the table and the similarity between the abstract and the keywords associated with the table.

3.3 Measuring the processing time

Measuring the processing time is an essential aspect when evaluating the performance of information retrieval systems and assessing their potential applicability in real-world scenarios. The increasing adoption of Artificial Intelligence models, especially in the field of NLP, has yielded impressive results. However, the execution of these models, which are becoming more sophisticated and resource-intensive, raises concerns about computational costs. Therefore, in this study, experiments were conducted to evaluate the execution time of various processes, aiming to determine the feasibility of implementing such systems in real-world scenarios.

The focus was on two key components: the offline data indexing component, which does not directly impact end-users, and the online classification process, which provides insights into the time required for classifying a single abstract. By assessing the execution time of these components, valuable insights are gained into the practicality of deploying these systems.

The experiments were conducted on a dataset consisting of over 20,000 abstracts, as described in Sect. 4.2. Through careful evaluation of the processing time it is possible to understand the performance characteristics and resource requirements of the system. This information is crucial for determining the system's feasibility and effectiveness in real-world applications.

4 Experiments

This section presents the experiments conducted to validate the hypothesis. The models used, the datasets collected, and the results obtained in each experiment are described in the following lines.

4.1 Models

The approach involves employing a diverse set of language models with various types of architectures, encompassing contextual, non-contextual, and fine-tuned models in the field of research. The objective is to compare the efficacy of these models in generating high-quality embeddings for matching abstracts with their corresponding tables in the scientific research field. For this purpose, the results of seven different models were compared:

1. Word2vec (Mikolov et al., 2013): non-contextual embedding vectors pre-trained using Google News dataset, comprising about 100 billion words. The model contains 300-dimensional vectors for 3 million words and phrases. These non-contextual model types are fast calculating the word embeddings but suffer from a limited vocabulary.
2. fastText (Bojanowski et al., 2016): non-contextual embedding vectors pre-trained on Wikipedia 2017, UMBC webbase corpus, and statmt.org news dataset, comprising about 16 billion words. As in the previous model, vectors have 300 dimensions. The pre-trained model containing subword information was used in these experiments to increase the coverage of the model's vocabulary.
3. WikiTables: non-contextual model specifically developed for this study. It uses skip-gram Word2vec trained on the Wikipedia Tables corpus, which contains 1.6 million Wikipedia relational tables (Bhagavatula et al., 2015). The corpus was pre-processed splitting CamelCase and hyphenated words, removing punctuation, and converting text to lowercase. For every table in this corpus, all the names of the columns were extracted and treated as an input document to train Word2vec.[2] A second model was created for the content of the cells. In this case, all the attribute values in a column were considered as an input document to train the model. Thus, there are two separate word embedding models to calculate the similarity between names of the columns and the content of the cells. The generated word embedding vectors are composed of 300 dimensions.
4. BERT (Devlin et al., 2019): developed by Google in 2018, it is a Transformer-based model pre-trained on a large corpus of English data in a self-supervised way. The model encoder is capable of generating contextual embedding vectors of 768 dimensions. BERT achieved state-of-the-art results on a wide range of NLP benchmarks, and its success has led to the development of many other language models based on its architecture.
5. SentenceBERT (Reimers & Gurevych, 2019): produces contextual embedding vectors using siamese and triplet network structures to derive semantically meaningful sentence embeddings. Specifically, the model `all-MiniLM-L6-v2` trained on 1 billion sentence pairs was used in our experiments. This model produces word embedding vectors of 768 dimensions.
6. RoBERTa (Liu et al., 2019): a BERT variant optimised at pre-training step. The encoder of this model also can be used to generate contextual embedding vectors. The model used in the experiments is `all-roberta-large-v1`, a large model fine-tuned with

[2] https://radimrehurek.com/gensim/models/word2vec.html.

1 billion of sentence pairs. This model produces an embedding vector of 1024 dimensions.
7. SciBERT (Beltagy et al., 2019): another variation of BERT specifically designed for scientific texts. It was pre-trained on a large corpus of the scientific documents comprising 1.14 million papers and 3.1 billion tokens from a variety of research fields. This model produces word embedding vectors of 768 dimensions.
8. BLOOM (Scao et al., 2022): BigScience Large Open-science Open-access Multilingual language model is another Transformer model based on Megatron-LM GPT2 and trained with 1.5 terabytes of pre-processed text, covering 350 billions of unique tokens to perform text generation tasks. It can produce coherent output in 46 languages and 13 programming languages. This model produces word embedding vectors of 1024 dimensions. In contrast to BERT-based models, this one does not have an encoder architecture but rather a decoder architecture.
9. SPECTER (Cohan et al., 2020): is a pre-trained language model that generates document-level word embeddings. It is pre-trained on the citation graph of scientific publications. Given the combination of title and abstract of a scientific paper or a short textual query, the model can be used to generate effective word embeddings to be used in downstream applications. This model was trained with 684,000 training triples and produces word embedding vectors of 768 dimensions.

This set of models comprises a diverse range of architectures, each with notable differences. One crucial distinction lies in how word embeddings are constructed and handled for out-of-vocabulary words. Traditional word-based models lack representations for out-of-vocabulary words since their embeddings are built on a fixed set of tokens present in the corpus. However, contextual models address this issue by employing subword segmentation algorithms like WordPiece, as mentioned in Sect. 2. These models initialise their vocabulary with individual characters and gradually add frequent combinations of symbols as subwords. This approach enables subwords to have their own representations in the semantic space, allowing previously unseen words to be represented by combining the vectors of their constituent subword units.

Furthermore, the models in this set differ in terms of their training data. Some models are trained on general-purpose text, such as Word2Vec, fastText, BERT, Sentence-BERT, and RoBERTa. Others are specifically trained on scientific texts, such as SciBERT, BLOOM and SPECTER. Additionally, WikiTables is specifically trained for table-related information.

4.2 Datasets

Before conducting the experiments, the acquisition of a suitable corpus consisting of research paper abstracts and their associated research data tables was necessary. Unfortunately, no corpus was found with these specific characteristics. Consequently, a methodology was proposed to merge two distinct corpora in order to gather the desired information. The following sections elaborate the merging process employed to obtain a corpus that satisfies the requirements of this study.

The first corpus utilised PubTabNet,[3] which encompasses a diverse array of tables comprising over 516,000 images containing tabular data. Each table is meticulously annotated with the corresponding HTML representation of its contents. PubTabNet primarily serves the purpose of training and evaluating image-based table recognition models. In the present work, tables were extracted from PubTabNet in HTML format, subsequently converting them to CSV format. Additionally, the PMID (unique identifier number employed in PubMed for each article) associated with each table was recorded, denoting the originating paper.

The subsequent objective involved linking these tables with their respective abstracts. This task was accomplished by leveraging the resources provided by MeSHup,[4] an extensively annotated corpus encompassing MeSH (Medical Subject Headings) indexing. MeSHup consists of 1,342,667 full-text papers, complete with associated MeSH labels and metadata. The corpus is represented as a large JSON file, which includes pertinent information such as the required abstracts, keyword sets, PMCID (unique identifier assigned to every article accepted into PubMed Central), and publication year.

By integrating the aforementioned datasets, correspondences between the abstracts, keywords, and their respective tables were established. It is important to note that due to disparities between the two datasets, matches were not available in many instances. However, 131,359 pairs of abstracts and tables were identified. To ensure computational efficiency, a filtering mechanism was applied to exclusively use articles published within the last five years. Ultimately, the resultant dataset comprised 23,744 pairs of abstracts and tables. A comprehensive summary of the dataset employed in the experiments is presented in Table 1.

4.3 Results

This section presents the evaluation of the performance of the aforementioned models in the classification task and examine the impact of integrating contextual information. The objective is to assess the effectiveness of these models and explore the potential benefits gained through the inclusion of contextual data.

4.3.1 Impact of varying α

The primary objective of this experiment is to evaluate the similarity between the embeddings of the abstracts and their corresponding tables, considering the variation of the parameter α. This evaluation enables to assess whether the header or the table content itself exhibits greater similarity with the abstract.

Figure 3 illustrates the results obtained for all the models: SentenceBERT (stb), BERT (brt), RoBERTa (rbt), SciBERT (sci), Word2vec (w2v), fastText (fst), BLOOM (blo), SPECTER (spe), and WikiTables (wiki). This figure demonstrates a slight increase in similarity as the α value increases across most models, giving prevalence to the headers, with the exception of BERT and SPECTER. Notably, the WikiTables model demonstrates higher similarity scores when the table headers are given more weight (higher α values).

[3] https://developer.ibm.com/exchanges/data/all/pubtabnet/.

[4] https://github.com/xdwang0726/MeSHup.

Table 1 Summary of the main characteristics of the dataset used in the experiments

Characteristic	Value
Number of tables	23,744
Number of rows	276,495
Number of columns	108,072
Number of numerical columns	21,878
Avg. abstract length (words)	243
Avg. number of rows	13
Avg. number of columns	5
Avg. number of numerical columns	1
Max. abstract length (words)	1172
Max. number of rows	71
Max. number of columns	39
Max. number of numerical columns	38

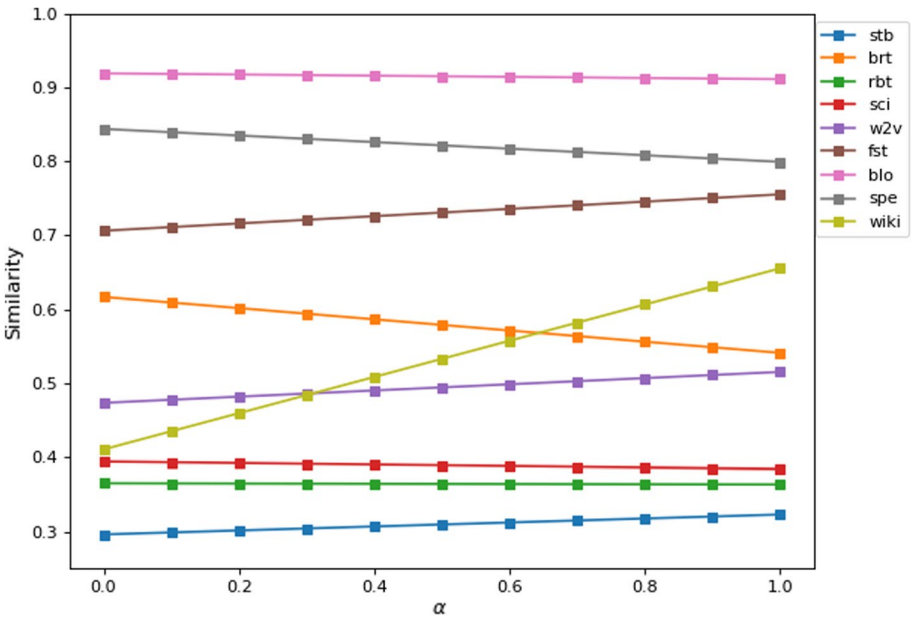


Fig. 3 Similarity obtained comparing tables with abstracts depending on the α parameter

It is important to note the divergent ranges of similarity scores obtained by each model. For instance, BLOOM and Specter produce relatively high similarity scores, averaging around 0.90, while others, such as SentenceBERT, remain closer to 0.30. This variation in similarity scores prompts further investigation into the impact of α variation on the models' performance and whether relying solely on abstract-table embeddings is sufficient for the intended task.

However, it is crucial to note that a high similarity score does not necessarily indicate true similarity between the abstract and tables. It could be influenced by factors

Table 2 MMR performance of the proposed models depending on α variations

Model	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
SentenceBERT	0.04	0.05	0.05	0.05	0.05	0.06	0.07	0.08	0.09	0.09	0.09
BERT	0.0	0.01	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.01	0.01
RoBERTa	0.05	0.05	0.06	0.06	0.06	0.06	0.07	0.08	0.09	0.09	0.09
SciBERT	0.04	0.04	0.05	0.05	0.05	0.06	0.06	0.08	0.10	0.10	0.10
Word2vec	0.0	0.01	0.0	0.0	0.0	0.01	0.01	0.01	0.01	0.01	0.01
fastText	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BLOOM	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SPECTER	0.04	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.07
WikiTables	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

such as non-uniform distribution within the vector space of certain models or the model training process not being specifically focused on semantic similarity tasks.

4.3.2 Table classification

In the second experiment, the performance of the models was evaluated by ranking the indexed tables based on their similarity to the corresponding abstracts. Two metrics, namely the mean reciprocal rank (MMR) and precision at different scales (Top@1, Top@3, and Top@5), were utilised to assess the performance of the models.

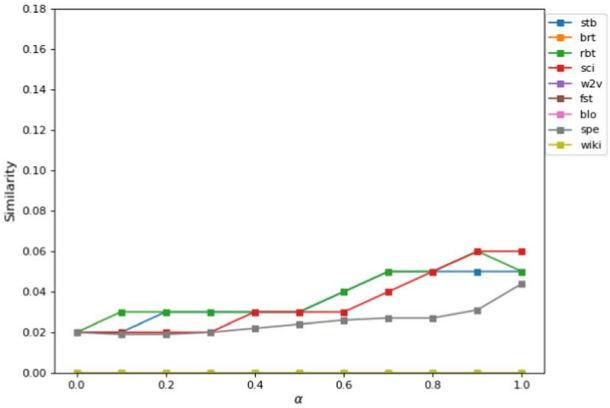
Table 2 presents the results obtained in terms of MMR, revealing that the overall performance in the task is not promising. However, the data does provide valuable insights into observable trends. Notably, there is a consistent pattern of improved performance across all models as the α values increase. Moreover, certain models, including SentenceBERT, RoBERTa, SPECTER, and SciBERT, exhibit notably superior performance compared to others. Conversely, the remaining models demonstrate relatively low performance metrics, approaching 0.

The performance of the models at various precision values (P@k) is depicted in Fig. 4. It is evident that certain models consistently outperform others across all values of k. However, no clear trend of performance improvement can be observed when increasing this value.

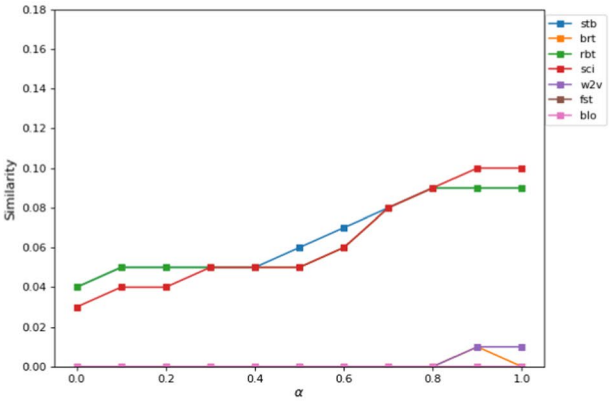
4.3.3 Impact of contextual information

The objective of the current experiment is to assess the impact of incorporating contextual information on the ranking performance when identifying the most similar tables to a given abstract. In this case, the keywords associated with the tables within the paper are considered as the context.

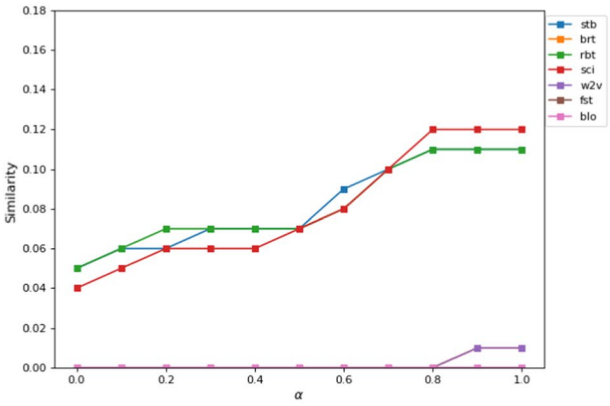
The results of applying the equations proposed in Sect. 3.2.1 are presented in Table 3. *Average* represents the approach described in Eq. 2 and *Boost* the approach described in Eq. 3. Two notable findings can be observed from the outcomes. Firstly, the approach that averages the similarities surpasses the initial approach, demonstrating performance gains that double the original scores for certain models. Secondly, the incorporation of contextual information proves to be crucial in achieving performance enhancements across all



(a) Precision@1



(b) Precision@3



(c) Precision@5

Fig. 4 Precision@k at different levels for each model varying α parameter

Table 3 MMR performance of the proposed models considering the two approaches that leverage context (*Average* and *Boost*)

Model	Average	Boost
SentenceBERT	0.27	0.23
BERT	0.09	0.03
RoBERTa	0.27	0.24
SciBERT	0.40	0.22
Word2vec	0.02	0.02
fastText	0.00	0.00
BLOOM	0.00	0.00
SPECTER	0.29	0.13
WikiTables	0.00	0.00

models, resulting in up to four times the performance improvement for SciBERT. It is also noteworthy that even with this approach, non-contextual models manage to obtain a significant performance boost.

Figure 5 illustrates the precision scores (P@K) for the two approaches, *Average* and *Boost*, as well as their evolution as k increases. Notably, SciBERT stands out as the most prominent model, consistently achieving high precision scores. At P@5, SciBERT demonstrates a precision score of nearly 0.5, indicating its strong performance in identifying relevant tables for a given abstract.

In addition, we conducted experiments with SciBERT ($\alpha = 0.8$), the best performing model, to investigate the impact of varying the β coefficient of Eq. 2 on the system's performance. The results in Fig. 6 reveal interesting findings regarding the performance at different precision levels (P@1, P@3, and P@5) when considering different values of β.

For P@1, the best performance is achieved with a β coefficient of 0.6. This suggests that for identifying the most similar table to a given abstract, a slightly higher weight should be given to the abstract-table pair rather than the abstract-context information.

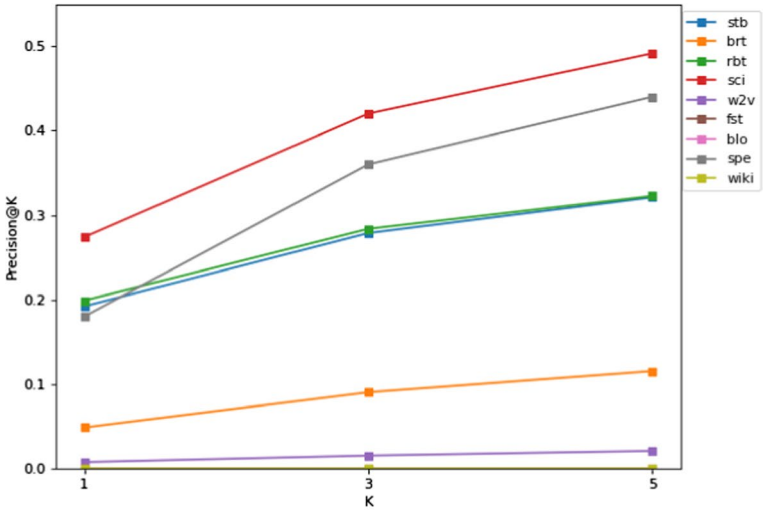
However, the results differ for P@3 and P@5, where a β coefficient of 0.8 demonstrates better performance. This performance significantly drops from this point on, when the coefficient is 0.9 and 1, and the weight of the context is almost (or totally) discarded.

These findings emphasise the importance of context in our system and highlight the need to balance the contribution of abstract and contextual information. By adjusting the β coefficient, the system can effectively leverage both sources of information to enhance its performance in retrieving relevant tables for a given abstract.

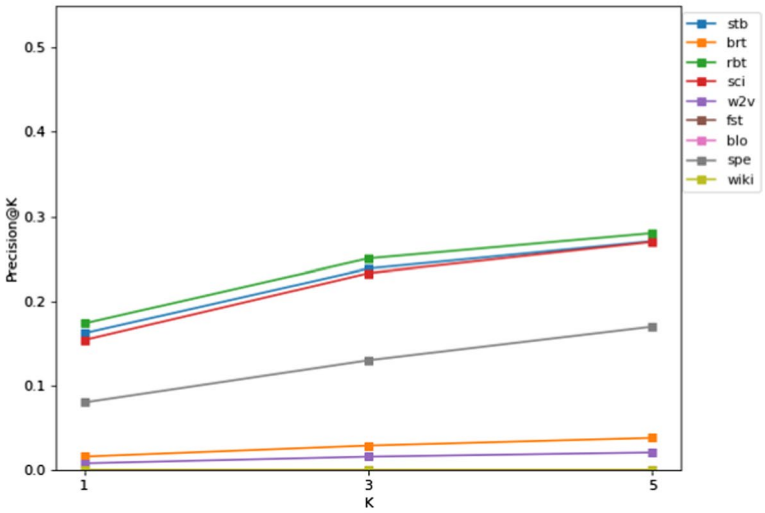
4.4 Execution time

The experiments were conducted using an A100-SXM4 GPU with 40 GB of RAM. Table 4 provides the execution times (in the format hours:minutes) for two aspects of interest during the experiments: indexing and classification time. It is worth noting that the average retrieval time for processing an abstract-table pair is approximately 150 milliseconds.

The total information indexing process for the models analysed took approximately 15 h. It is important to note that the computational intensity varied across different models, impacting the speed of embedding calculation. Among the models used, non-contextual models such as Word2vec and fastText exhibited faster computation times compared to



(a) Average



(b) Boost

Fig. 5 Precision@k at different levels for the two approaches that leverage context (*Average* and *Boost*)

the other models. These models were able to generate word embeddings relatively quickly, contributing to the efficiency of the indexing process.

On the other hand, the BLOOM model was the slowest in terms of computation time. Despite its slower performance, it should be noted that the effectiveness of BLOOM in producing satisfactory results for the experimentation presented in this study was lower compared to other models.

Additionally, the classification processes and metric calculation took approximately 28 h in total. These tasks involved evaluating the performance of the models and calculating

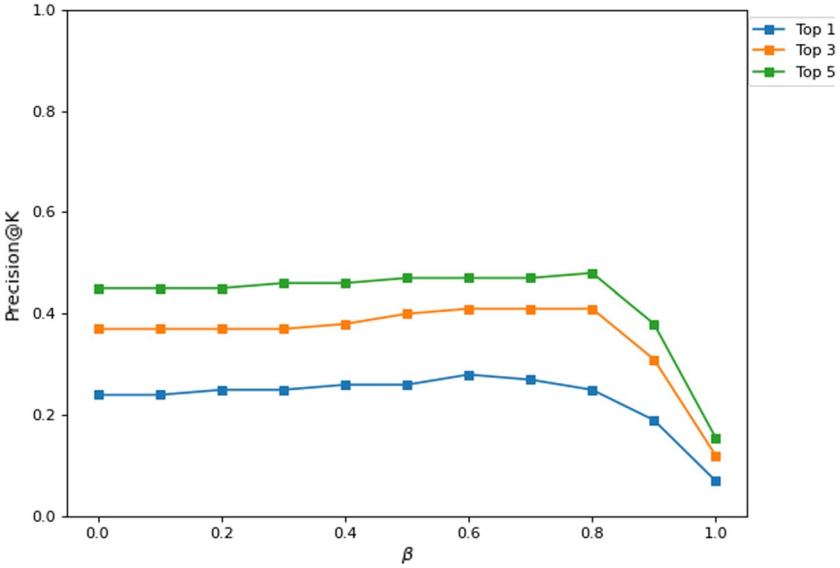


Fig. 6 SciBERT precision@k varying β coefficient in Eq. 2

Table 4 Indexing and classification time (hours:minutes) for each model

Model	Indexing	Classification
SentenceBERT	01:07	03:18
BERT	01:31	03:52
RoBERTa	02:15	04:34
SciBERT	01:36	03:47
Word2vec	00:46	03:17
fastText	00:52	03:17
BLOOM	04:16	05:23
SPECTER	03:38	04:26
WikiTables	01:51	04:32

metrics such as MMR and precision at different scales (P@1, P@3, and P@5) for all the 23,744 pairs of abstracts and tables.

The previous experiments were conducted to achieve comprehensive research results rather than focusing on real-time production systems. However, it is possible to adapt the proposed approach to improve response times in a real-time setting. By implementing certain optimisations, the system can be made more efficient without compromising its overall performance.

One such optimisation is to utilise the GPU version of Faiss, which leverages the computational power of GPUs, resulting in significant improvements in embedding search speed. According to the official Faiss documentation, using the GPU version can lead to speed improvements of up to 5–10 times compared to the CPU version, providing faster computation and reduced response times.

Furthermore, Faiss provides various techniques that can further optimise the search process. For instance, partitioning of the index can be applied to reduce the search range and

enhance efficiency. This technique involves dividing the index into smaller subsets, enabling faster search operations by narrowing down the search space. By implementing partitioning, the system can achieve faster search times without sacrificing accuracy. Another optimisation technique is vector quantisation, which involves reducing the dimensionality of the vectors. By mapping the original vectors to a set of predefined centroids and storing only the indices of these centroids, the vector size can be reduced.

It is important to note that while these optimisation techniques can enhance system speed, there may be a trade-off with system performance. For example, the accuracy of search results may be slightly affected, and the choice of partitioning strategies or the number of centroids for vector quantisation can impact the overall system performance. Therefore, it is crucial to carefully evaluate the trade-offs between system performance and speed to determine the optimal configuration for the specific real-time use case.

5 Discussion

The thorough examination of the experimental results led to the hypothesis that accurate preprocessing of the data is crucial for achieving reliable outcomes. Various issues were identified during the analysis, including malformed tables, multi-header structures, non-standard codification, and the need for data transformation, such as data tidying.

Malformed tables can introduce inconsistencies and irregularities in the data, making it challenging for the models to extract meaningful information. Multi-header structures, where tables have multiple layers of headers, can confuse the models and affect their ability to understand the table's structure and content. Non-standard codification or encoding issues can lead to misinterpretation of the data, causing inaccuracies in the embeddings and subsequent similarity calculations. Additionally, certain tables may require data tidying operations, such as handling missing values, standardising units, or normalising data, to ensure consistent and reliable representations.

Furthermore, the use of broad and general keywords, such as `USA`, `Biology`, or `Male`, may not provide significant information when used as context in this specific experiment. Since all the documents are highly related to the medical field, these keywords may not contribute much to distinguishing the tables' relevance or similarity to the abstracts.

Numeric columns pose another challenge. The percentage of these columns is not very large, but they can contain various formats such as percentages, ranges, or scientific notation. While some models are capable of handling numerical data, associating and finding similarity between numeric values and abstracts or sets of keywords can be difficult. This can impact the performance of the approach, particularly in cases where the table's numerical information is essential for determining its relevance.

The results highlight the significance of context in assessing the relevance of a table. Tables without context can be ambiguous, making it difficult to understand their purpose and the type of information they contain. For example, a table with columns indicating day and temperature may have different interpretations without proper context. Contextual information is crucial for disambiguating the purpose and meaning of tables, whether they represent temperatures in a specific location, a patient's body temperature, or the Earth's core temperature. The absence of clear context can pose challenges to the proposed approach.

6 Conclusions and future work

This paper presents a study that explores the impact of different language models on the retrieval of tables from research publication abstracts. The primary objective is to demonstrate the feasibility of extracting semantic information from both the abstracts and tables to facilitate information retrieval. The study also investigates the influence of incorporating contextual information in the task.

The results obtained emphasise the importance of high-quality data and preprocessing steps in improving the performance of the retrieval task. Specifically, the inclusion of contextual information, represented by keywords, proves to be crucial. For example, when used with SentenceBERT, the incorporation of keywords triples the performance in the task.

Contextual models outperform non-contextual models significantly, showcasing their superior performance. However, it is important to note that these models exhibit higher computational complexity, leading to slower execution times. Furthermore, the study highlights the substantial improvement achieved by BERT models trained on scientific data compared to the base BERT model, underscoring the impact of pre-training data on model performance. The experimental findings indicate that contextual models utilising encoder architecture outperform BLOOM, which employs a decoder architecture resulting in lower-quality embeddings.

Several avenues for future research are identified based on this study. First, there is a need to explore contextual models with encoder architecture specifically fine-tuned for the abstract-table comparison task. Additionally, addressing the challenges posed by numeric columns and improving data preprocessing to handle higher quality data are important areas for future work.

Furthermore, future research should investigate the extent to which additional contextual information impacts classification accuracy. One potential area of exploration is the inclusion of table captions as a valuable resource to enhance the embedding representation of the dataset. By incorporating table captions, which often provide concise and informative summaries of the table content, the accuracy of the classification task may be further improved.

Finally, the authors envision the development of a classification software based on their approach, aiming to support scientists in finding relevant datasets related to their research problems.

Author Contributions The authors contributed to this work as follows: AB: Methodology, Data Curation, Software, Investigation, Writing - Original Draft DT: Conceptualization, Supervision, Methodology, Formal Analysis, Software, Investigation, Writing - Original J-NM: Conceptualization, Supervision, Resources, Writing - Original Draft

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work is part of the project TED2021-130890B-C21, funded by MCIN/AEI/10.1 3039501100011033 and by the European Union NextGenerationEU/PRTR. Alberto Berenguer has a contract for predoctoral training with the Generalitat Valenciana and the European Social Fund, funded by the grant ACIF/2021/507.

Data availability The datasets used in the experiments are available in the following URLs:
MeSHup: https://github.com/xdwang0726/MeSHup
PubTabNet: https://developer.ibm.com/exchanges/data/all/pubtabnet/

Code availability The code is freely available in a GitHub repository: https://github.com/aberenguerpas/abstract-dataset-pairing

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Agarwal, V., Bhardwaj, A., Rosso, P., & Cudré-Mauroux, P. (2021). Convtab: A context-preserving, convolutional model for ad-hoc table retrieval. In *2021 IEEE International Conference on Big Data (Big Data)*, pp. 5043–5052, https://doi.org/10.1109/BigData52589.2021.9671828

Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, Hong Kong, China, pp. 3615–3620, https://doi.org/10.18653/v1/D19-1371. https://aclanthology.org/D19-1371

Bhagavatula, C.S., Noraset, T., & Downey, D. (2013). Methods for exploring and mining tables on wikipedia. In *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*, pp. 18–26

Bhagavatula, C.S., Noraset, T., & Downey, D. (2015). Tabel: Entity linking in web tables. In *The Semantic Web - ISWC 2015*, Springer International Publishing, Cham, pp. 425-441.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching word vectors with subword information. CoRR arXiv:1607.04606.

Cafarella, M. J., Halevy, A. Y., Wang, D. Z., Wu, E., & Zhang, Y. (2008). Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment, 1*(1), 538–549.

Cafarella, M. J., Halevy, A. Y., & Khoussainova, N. (2009). Data integration for the relational web. *Proceedings of the VLDB Endowment, 2*(1), 1090–1101.

Chen, J., Wang, X., Cheng, G., Kharlamov, E., & Qu, Y. (2019). Towards more usable dataset search: From query characterization to snippet generation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 2445–2448.

Chen, Z., Trabelsi, M., Heflin, J., Xu, Y., & Davison, B.D. (2020). Table search using a deep contextualized language model. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Association for Computing Machinery, Virtual, pp. 589–598, https://doi.org/10.1145/3397271.3401044

Cohan, A., Feldman, S., Beltagy, I., Downey, D., & Weld, D.S. (2020). Specter: Document-level representation learning using citation-informed transformers. arXiv:2004.07180.

Deng, L., Zhang, S., & Balog, K. (2019). Table2vec: Neural word and entity embeddings for table population and retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Association for Computing Machinery, Paris, France, pp. 1029–1032, https://doi.org/10.1145/3331184.3331333

Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 4171–4186, https://doi.org/10.18653/v1/N19-1423

Du, L., Gao, F., Chen, X., Jia, R., Wang, J., Jiang, Z., Han, S., & Zhang, D. (2021). Tabularnet: A neural network architecture for understanding semantic structures of tabular data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery*, New York, NY, USA, KDD '21, pp 322—331, https://doi.org/10.1145/3447548.3467228

Färber, M., & Leisinger, A.K. (2021). Recommending datasets for scientific problem descriptions. In CIKM, pp. 3014–3018.

Firth, J.R. (1957). A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*, Blackwell, Oxford, pp. 1-32.

Gregory, K. M., Cousijn, H., Groth, P., Scharnhorst, A., & Wyatt, S. (2020). Understanding data search as a socio-technical practice. *Journal of Information Science, 46*(4), 459–475.

Hagiwara, Y., Ishita, E., Watanabe, Y., & Tomiura, Y. (2022). Identifying scholarly search skills based on resource and document selection behavior among researchers and master's students in engineering. *College & Research Libraries, 83*(4), 610.

Harris, Z. S. (1954). Distributional structure. *Word, 10*(2–3), 146–162. https://doi.org/10.1080/00437956.1954.11659520

Herzig, J., Nowak, P.K., Müller, T., Piccinno, F., & Eisenschlos, J. (2020). TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Virtual, pp. 4320–4333, https://doi.org/10.18653/v1/2020.acl-main.398

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics*, Melbourne, Australia, pp. 328–339.

Kang, H. B., Qian, X., Hope, T., Shahaf, D., Chan, J., & Kittur, A. (2022). Augmenting scientific creativity with an analogical search engine. *ACM Transactions on Computer-Human Interaction, 29*(6), 1–36.

Lilleberg, J., Zhu, Y., & Zhang, Y. (2015). Support vector machines and word2vec for text classification with semantic features. In *2015 IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, pp. 136–140.

Liu, Q., Kusner, M.J., & Blunsom, P. (2020). A Survey on Contextual Embeddings. arXiv preprint arXiv:2003.07278.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. CoRR arXiv:1907.11692.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - vol. 2*, Curran Associates Inc., Lake Tahoe, Nevada, NIPS'13, pp. 3111–3119.

Mysore, S., Jasim, M., Song, H., Akbar, S., Randall, A.K.C., & Mahyar, N. (2023). How Data Scientists Review the Scholarly Literature. arXiv preprint arXiv:2301.03774.

Paullada, A., Raji, I. D., Bender, E. M., Denton, E., & Hanna, A. (2021). Data and its (dis) contents: A survey of dataset development and use in machine learning research. *Patterns, 2*(11), 100336.

Pennington, J., Socher, R., & Manning, C.D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, http://www.aclweb.org/anthology/D14-1162.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational inguistics: Human Language Technologies*, Vol. 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, pp. 2227–2237, https://doi.org/10.18653/v1/N18-1202. https://www.aclweb.org/anthology/N18-1202.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. CoRR arXiv:1908.10084.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management, 24*(5), 513–523.

Scao, T.L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A.S., Yvon, F., Gallé, M., et al. (2022). Bloom: A 176b-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100.

Shih, K.J., Singh, S., & Hoiem, D. (2016). Where to look: Focus regions for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research, 37*, 141–188.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, Curran Associates Inc, Long Beach, CA, USA, 30*, 5998–6008.

Viswanathan, V., Gao, L., Wu, T., Liu, P., & Neubig, G. (2023). Datafinder: Scientific dataset recommendation from natural language descriptions. arXiv preprint arXiv:2305.16636.

Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., & Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. CoRR arXiv:1609.08144.

Yin, P., Neubig, G., tau, Yih, W., & Riedel, S. (2020). TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Virtual, pp. 8413–8426, https://doi.org/10.18653/v1/2020.acl-main.745

Zhang, L., Zhang, S., & Balog, K. (2019). Table2vec: Neural word and entity embeddings for table population and retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information retrieval*, pp. 1029–1032.

Zhang, S., & Balog, K. (2018). Ad hoc table retrieval using semantic similarity. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, ACM Press, pp. 1553–1562, https://doi.org/10.1145/3178876.3186067

Zhang, S., & Balog, K. (2020). Web table extraction, retrieval, and augmentation: A survey. *ACM Transactions on Intelligent Systems and Technology, 11*(2), 1–35. https://doi.org/10.1145/3372117